

**МИНИСТЕРСТВО ПО РАЗВИТИЮ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
КОММУНИКАЦИЙ РЕСПУБЛИКИ УЗБЕКИСТАН**

**НУКУССКИЙ ФИЛИАЛ ТАШКЕНТСКОГО УНИВЕРСИТЕТА ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**



ФАКУЛЬТЕТ: Компьютерный инжиниринг

КУРСОВАЯ РАБОТА

**По предмету: «Разработка программного обеспечения
встроенных систем»**

На тему: «Создание системы управления по Bluetooth и Wi-Fi»

Выполнила:

Тилепова А.

Принял:

Татлымуратов Н.

НУКУС 2017

ПЛАН:

Введение

1. Анализ беспроводных сетей

Беспроводная связь технологии Wi- Fi

Технология ближней беспроводной радиосвязи Bluetooth

2. Использование технологии BLUETOOTH при построении распределённых управляющих систем

Практическая часть

3.Связываем Arduino и Android через Bluetooth

Заключение

Список литературы

ВВЕДЕНИЕ

Исследование технологий сетей является очень важным в современных условиях рыночной экономики, в особенности в сфере информационных технологий. Так как на выбор сети оказывают влияние много факторов, наиболее важные из них: расстояние, качество связи, кодек и контейнер в котором закодирован файл. Приложение позволяет рассчитать приблизительное время передачи данных, так как заявленная в сети скорость не всегда является фактической. Таким образом можно подобрать оптимальную технологию для передачи данных в нужных условиях.

Беспроводная связь в первую очередь -возможность передачи информации на расстояния без кабельной системы. Преимущество беспроводной связи — простота установки. Когда не требуется прокладывать физические провода до офиса, процедура установки может быть быстрой и экономически эффективной. Беспроводная связь упрощает также подключение труднодоступных объектов, таких как складские и заводские помещения. Затраты на построение беспроводной связи обходятся дешевле, поскольку при этом ликвидируются проблемы с организацией прокладки проводов и затраты, связанные с этим процессом. В беспроводной связи наиболее распространенными и известными на сегодняшний день являются три семейства технологий передачи информации, такие как Wi-Fi, GSM, Bluetooth. Эти технологии детально рассматриваются в плане защищенности от возможных атак.

Технология *Wi-Fi* (сокращение от wireless fidelity — «Беспроводная надёжность») применяется при соединении большого количества компьютеров. Другими словами, это беспроводное подключение к сети. Одна из самых перспективных технологий на сегодняшний день в области компьютерной связи.

GSM- (Global System for Mobile Communications — глобальная система связи с подвижными объектами). Технология GSM родилась в недрах компании Group Special Mobile, от которой и получила сокращение GSM, однако со временем сокращение получило другую расшифровку Global System for Mobile.

Bluetooth -это технология беспроводной передачи данных малой мощности, разрабатываемая с целью замены существующих проводных соединений персональных офисной и бытовой техники с широким спектром переносных устройств, таких, как мобильные телефоны и гарнитуры к мобильным телефонам, датчики сигнализации и телеметрии, электронные записные книжки и карманные компьютеры.

1. АНАЛИЗ БЕЗПРОВОДНЫХ СЕТЕЙ

1.1 Беспроводная связь технологии Wi-Fi

Wi-Fi — это современная технология беспроводного доступа в интернет. Доступ в интернет по технологии Wi-Fi осуществляется посредством специальных радиоточек доступа.(AP Access Point).

Существует такие разновидности Wi — Fi сетей:

801.11a;

801.11b;

801.11g;

801.11n.

Первая работает на частоте 5 ГГц, остальные на частоте 2.4ГГц. Каждый тип имеет разную пропускную способность (максимально теоретически возможную скорость):

для 801.11a это 54 Мбит/с;

для 801.11b это 11 Мбит/с;

для 801.11g это 54 Мбит/с;

для 801.11n это 600 Мбит/с.

Любая беспроводная сеть состоит как минимум из двух базовых компонентов — точки беспроводного доступа, клиента беспроводной сети (режим ad-hoc, при котором клиенты беспроводной сети общаются друг с другом напрямую без участия точки доступа). Стандартами беспроводных сетей 802.11a/b/g предусматривается несколько механизмов обеспечения безопасности, к которым относятся различные механизмы аутентификации пользователей и реализация шифрования при передаче данных. На сегодняшний день практически все современные портативные и карманные компьютеры являются Wi-Fi-совместимыми.

Преимущества Wi-Fi:

- простой и удобный способ подключения к услуге;

- отсутствие необходимости подключения дополнительных устройств — модемов, телефонных линий, выделенных каналов для соединения с сетью Интернет;
- простой способ настройки компьютера;
- нет зависимости от времени использования услуги, оплата только за используемый Интернет — трафик;
- скорость приема/передачи данных — до 54 Мбит/с;
- защищенность передачи данных;
- постоянно расширяемая сеть точек доступа Wi-Fi.

Рассмотрим недостатки Wi-Fi. Частотный диапазон и эксплуатационные ограничения в различных странах неодинаковы. Во многих европейских странах разрешены два дополнительных канала, которые запрещены в США; В Японии есть ещё один канал в верхней части диапазона, а другие страны, например Испания, запрещают использование низкочастотных каналов.

1.1.1 Описание протоколов безопасности беспроводной сети Wi — Fi

Все современные беспроводные устройства (точки доступа, беспроводные адаптеры и маршрутизаторы) поддерживают протокол безопасности WEP (Wired Equivalent Privacy), который был изначально заложен в спецификацию беспроводных сетей IEEE 802.11. Протокол WEP используется для обеспечения конфиденциальности и защиты передаваемых данных авторизированных пользователей беспроводной сети от прослушивания. Существует две разновидности WEP: WEP-40 и WEP-104, различаются только длиной ключа. В настоящее время данная технология является устаревшей, так как ее взлом может быть осуществлен всего за несколько минут. Тем не менее, она продолжает широко использоваться. Для безопасности в сетях Wi-Fi рекомендуется использовать WPA.

Процедура WEP-шифрования выглядит следующим образом: первоначально передаваемые в пакете данные проверяются на целостность (алгоритм CRC-32), после чего контрольная сумма (integrity check value, ICV) добавляется в служебное поле заголовка пакета. Далее генерируется 24-битный вектор инициализации, (IV) и к нему добавляется статический (40-или 104-битный) секретный ключ. Полученный таким образом 64-или 128-битный ключ и является исходным ключом для генерации псевдослучайного числа, используемого для шифрования данных. Далее данные шифруются с помощью логической операции XOR с псевдослучайной ключевой

последовательностью, а вектор инициализации добавляется в служебное поле кадра .

Кадр WEP включает в себя следующие поля:

- незашифрованная часть;
- вектор инициализации (англ. Initialization Vector) (24 бита);
- пустое место (англ. Pad) (6 бит);
- идентификатор ключа (англ. Key ID) (2 бита);
- зашифрованная часть;
- данные;
- контрольная сумма (32 бита).

1.2 Технология ближней беспроводной радиосвязи Bluetooth

Технология Bluetooth получила свое название в честь датского короля X-го века Гаральда II Блатана. В переводе с датского «Блатан» — Синий Зуб, соответственно в английском варианте — Bluetooth. Этот король прославился своей способностью находить общий язык с князьями-вассалами и в свое время объединил Данию и Норвегию. Через 1000 лет его имя предложила в качестве названия для новой технологии шведская компания Ericsson, которая выступила инициатором проекта Bluetooth. Bluetooth _ технология беспроводной передачи данных, позволяющая соединять друг с другом любые устройства, в которых имеется встроенный микрочип Bluetooth. Наиболее активно технология применяется для подключения к мобильным телефонам всевозможных внешних устройств: беспроводных гарнитур handsfree, беспроводных модемов, приемников спутниковой навигации, и собственно для подключения к персональному компьютеру.

Bluetooth может общаться с несколькими (до семи) устройствами Bluetooth: одно устройство при этом будет активным, а остальные находятся в режиме ожидания. Радиоволны, которые используются в Bluetooth, могут проходить через стены и неметаллические барьеры и соединяться с Bluetooth-устройствами на расстоянии от 10 до 100 метров в зависимости от спецификации устройства. Для спецификации 1.1 класс 1 радиус действия составляет до 100 метров, для класса 2 (применяемого в мобильных телефонах) _ до 10 — м. Так как во всем мире Bluetooth работает на не лицензируемой и единой частоте промышленного, научного и медицинского применения ISM 2,45 ГГц, то пространственных границ для использования

Bluetooth не существует. Как не существует и проблемы несовместимости Bluetooth-устройств различных производителей, поскольку технология стандартизирована. Так что никаких препятствий для распространения Bluetooth нет. Каждое Bluetooth устройство имеет свой уникальный адрес и имя, поэтому после процедуры регистрации соединяется только с зарегистрированным с ним телефоном. Для настройки необходимо зарядить гарнитуру, включить оба устройства (телефон и гарнитуру) и поместить поблизости друг от друга. После запуска процедуры поиска гарнитуры на дисплее телефона высветится ее спецификация и будет запрошен пароль (обычно требуется ввести пароль 0000). После его введения гарнитура считается зарегистрированной за вашим телефоном. Однако при всех плюсах Bluetooth, есть у него 3 огромных минуса: невысокая дальность действия, низкая (в сравнении с тем же Wi-Fi) скорость и огромное количество мелких и не очень «ошибок». И если с первыми двумя недостатками можно мириться или бороться, то количество недоработок заставляет поразиться любого, даже далёкого от высоких технологий человека.

1.2.1 Спецификации Bluetooth

Устройства версий 1.0 (1998) и 1.0B, имели плохую совместимость между продуктами различных производителей. В 1.0 и 1.0B была обязательной передача адреса устройства (BD_ADDR) на этапе установления связи. И делало невозможной реализацию анонимности соединения на протокольном уровне и, было основным недостатком данной спецификации.

Bluetooth 1.1 было исправлено множество ошибок, найденных в 1.0B, добавлена поддержка для нешифрованных каналов, индикация уровня мощности принимаемого сигнала (RSSI). В версии 1.2 была добавлена технология адаптивной перестройки рабочей частоты (AFH), что улучшило сопротивляемость к электромагнитной интерференции (помехам) путём использования разнесённых частот в последовательности перестройки. Также увеличилась скорость передачи и добавилась технология eSCO, которая улучшала качество передачи голоса путём повторения повреждённых пакетов. В HCI добавилась поддержка трёх-проводного интерфейса UART.

Главные улучшения включают следующее:

- быстрое подключение и обнаружение;
- адаптивная перестройки частоты с расширенным спектром (AFH), которая повышает стойкость к радиопомехам;
- более высокие, чем в 1.1, скорости передачи данных, практически до 721 кбит/с;

-расширенные Синхронные Подключения (eSCO), которые улучшают качество передачи голоса в аудио потоке, позволяя повторную передачу повреждённых пакетов, и при необходимости могут увеличить задержку аудио, чтобы оказать лучшую поддержку для параллельной передачи данных.

Bluetooth версии 2.0 был выпущен 10 ноября 2004 г. Имеет обратную совместимость с предыдущими версиями 1.x. Основным нововведением стала поддержка EDR (Enhanced Data Rate) для ускорения передачи данных. Номинальная скорость EDR около 3 Мбит/с, однако, на практике это позволило повысить скорость передачи данных только до 2,1 Мбит/с. Дополнительная производительность достигается с помощью различных радио технологий для передачи данных. Bluetooth SIG издала спецификацию как «Технология Bluetooth 2.0 + EDR», которая подразумевает, что EDR является дополнительной функцией. Согласно 2.0 + EDR спецификации, EDR обеспечивает следующие преимущества:

увеличение скорости передачи в 3 раза (2,1 Мбит/с) в некоторых случаях;

уменьшение сложности нескольких одновременных подключений из-за дополнительной полосы пропускания;

более низкое потребление энергии благодаря уменьшению нагрузки.

Bluetooth 3.0 + HS спецификация была принята Bluetooth SIG 21 апреля 2009 года. Она поддерживает теоретическую скорость передачи данных до 24 Мбит/с. Её основной особенностью является добавление AMP (Асимметричная Мультипроцессорная Обработка) (альтернативно MAC/PHY), дополнение к 802.11 как высокоскоростное сообщение. Две технологии были предусмотрены для AMP: 802.11 и UWB, но UWB отсутствует в спецификации.

Модули с поддержкой новой спецификации соединяют в себе две радиосистемы: первая обеспечивает передачу данных в 3 Мбит/с (стандартная для Bluetooth 2.0) и имеет низкое энергопотребление; вторая совместима со стандартом 802.11 и обеспечивает возможность передачи данных со скоростью до 24 Мбит/с (сравнима со скоростью сетей Wi-Fi). Выбор радиосистемы для передачи данных зависит от размера передаваемого файла. Небольшие файлы передаются по медленному каналу, а большие _ по высокоскоростному. Bluetooth 3.0 использует более общий стандарт 802.11 (без суффикса), то есть не совместим с такими спецификациями Wi-Fi, как 802.11b/g или 802.11n.

Bluetooth 4.0 пропускная способность осталась на уровне Bluetooth 3.0 со значением 24 Мбит/с, но дальность действия повысилась до 100 метров.

Одновременно с этим произошло снижение энергопотребления, что позволяет использовать технологию в устройствах на батарейках. Разработка также поддерживает шифрование AES-128 и предоставляет еще более низкое время отклика, повышая безопасность и становясь более удобной для пользователей.

1.2.2 Механизмы безопасности Bluetooth

Спецификация BT основана на модели обеспечения безопасности, предусматривающей три механизма: аутентификация (опознавание), авторизация (разрешение доступа) и шифрование (кодирование). Суть опознавания состоит в том, чтобы удостовериться, является ли устройство, инициирующее сеанс связи, тем, за кого оно себя выдает.

2. Использование технологии BLUETOOTH при построении распределённых управляющих систем

При выборе беспроводной технологии для проектируемой системы одним из главных требований является открытость стандарта, совместимость устройств различных производителей, доступность таких устройств в длительной перспективе, надежность и отработанность технологии, низкая цена, экономичность, возможность построения сетей различной архитектуры. Из всех существующих сегодня технологий в наибольшей степени этим требованиям соответствуют ZigBee (IEEE 802.15.4), WLAN (Wireless Local Area Network - беспроводная локальная вычислительная сеть. IEEE 802.11) и Bluetooth (IEEE 802.15.1). WLAN обеспечивает высокие скорости передачи информации (в зависимости от версии—11 или 54 Мбит/с) и, очевидно, предпочтительнее в системах, базирующихся на сетях Ethernet. Однако этот стандарт отличается повышенным энергопотреблением, да и цена на модули заметно выше, чем у ZigBee и Bluetooth. Основная особенность технологии ZigBee заключается в том, что она поддерживает не только простые топологии беспроводной связи («точка–точка» и «звезда»), но и сложные беспроводные сети с ретрансляцией и маршрутизацией сообщений с топологиями «дерево» или «ячеистая сеть». Модули ZigBee из всех перечисленных технологий отличаются самым низким энергопотреблением, однако скорость передачи информации в этих сетях не превышает 250 кбит/с. Области применения данной технологии -беспроводные сети датчиков, системы автоматизации зданий, системы управления в промышленности.

Bluetooth по скорости передачи информации и энергопотреблению занимает промежуточное положение между ZigBee и WLAN. Сейчас это самая популярная радио технология ближнего радиуса действия. По радиусу действия и назначению современные беспроводные сети можно разделить на персональные (Wireless Personal Area Network, WPAN),

локальные (Wireless Local Area Network, WLAN),

городские (Wireless Metropolitan Area Network, WMAN) и

глобальные (Wireless Wide Area Network, WWAN).

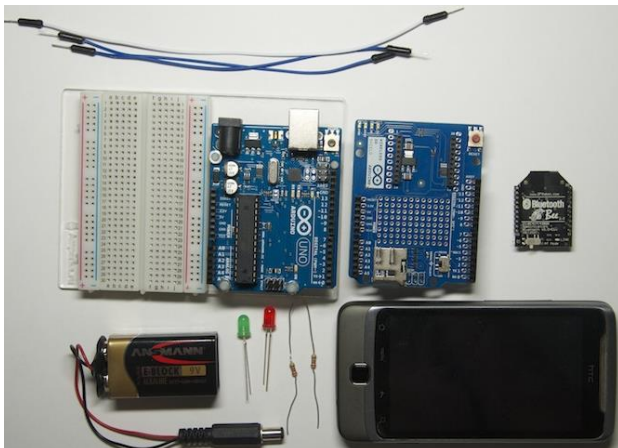
Для беспроводных решений Bluetooth производятся различные Bluetooth-модули. В отличие от ZigBee-модулей, Bluetooth-решения не предназначены для многомесячной автономной работы от батарей. Bluetooth-модули могут передавать как голосовую информацию, так и цифровые данные на расстояние 10-100 метров. Благодаря наличию стандартных интерфейсов, Bluetooth-модули могут легко интегрироваться в приборы различного назначения. Практически все Bluetooth-модули поддерживают профиль последовательного порта и могут использоваться для замены кабеля RS-232. При этом существующее программное обеспечение устройства практически не потребует никакой модернизации. Система Bluetooth предоставляет услуги по соединениям типа «точка-точка» для двух устройств Bluetooth или «точка-много точек». В последнем случае устройства Bluetooth способны соединяться друг с другом, формируя пикосети, когда одно из устройств является ведущим (Master), еще семь - ведомыми (Slave). Несколько пикосетей могут перекрываться, образуя распределенную сеть (scatternet) с общим числом устройств до 256. В момент присоединения к пикосети каждое Slave-устройство получает от мастера пакет, в котором содержится идентификатор Global ID, используемый для определения номера последовательности перестройки частоты, а также трехбитный адрес AMA (Active Member Address) для общения с соседями. Связь между модулем и хост-контроллером производится с помощью высокоскоростного USB-интерфейса или UART (Universal Asynchronous Receiver/Transmitter - универсальный асинхронный приёмопередатчик) / PCM (Pulse Code Modulation – импульсно-кодовая модуляция) - интерфейса. Хост направляет команды, а в ответ принимает от модуля сообщения об их выполнении. Менеджер связи устанавливает необходимую конфигурацию данного интерфейса .

Практическая часть

3.Связываем Arduino и Android через Bluetooth

В данной практической части будет подробно расписано создание небольшого приложения для мобильной операционной системы Android и скетча для Arduino. На Arduino Uno будет стоять Wireless Shield с Bluetooth-модулем. Приложение будет подключаться к Bluetooth-модулю и посылать некую команду. В свою очередь скетч по этой команде будет зажигать или гасить один из подключенных к Arduino светодиодов.

Нам понадобится

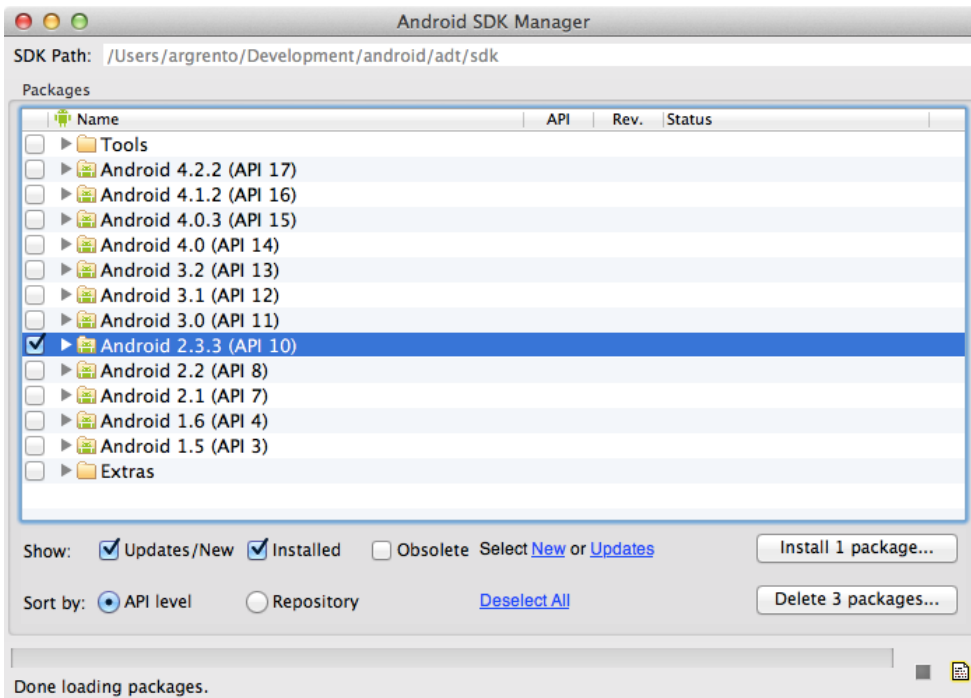


1. Arduino Uno
2. Arduino Wireless Shield
3. Bluetooth Bee
4. Красные и зеленые светодиоды
5. Резисторы
6. Провода «папа-папа»
7. Breadboard Half
8. Телефон на базе ОС Android

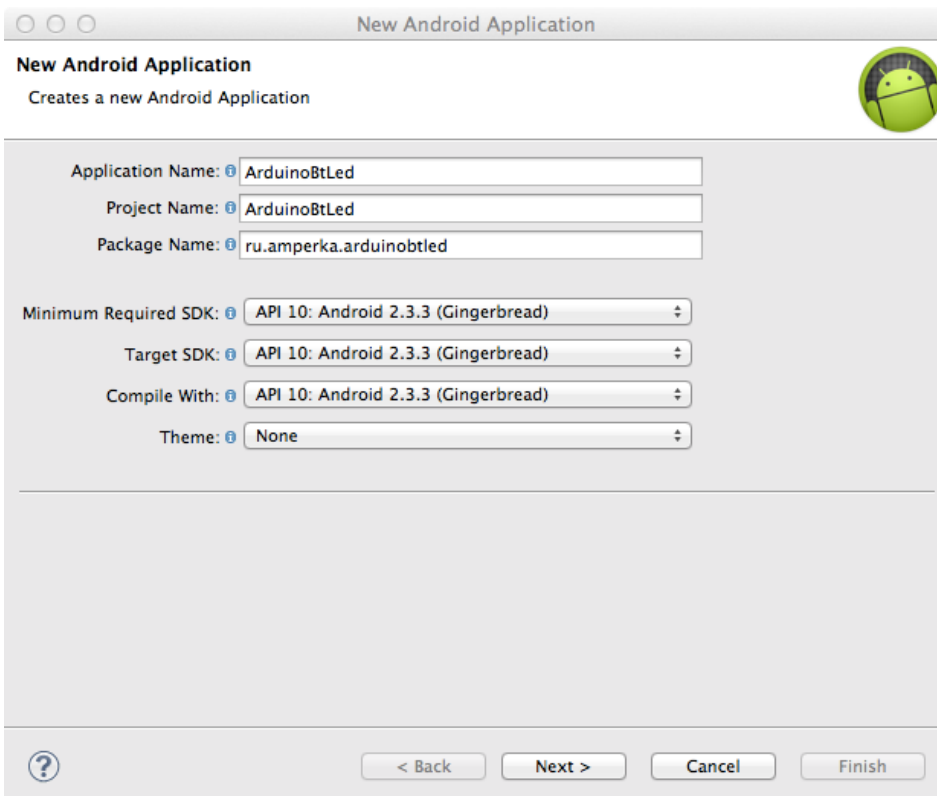
Создание приложения для Android

Заготовка

Разработка для ОС Android ведется в среде разработки ADT, Android Development Tools. Которую можно скачать с [портала Google](#) для разработчиков. После скачивания и установке ADT, смело его запускаем. Однако, еще рано приступать к разработке приложения. Надо еще скачать Android SDK нужной версии. Для этого необходимо открыть Android SDK Manager «Window → Android SDK Manager». В списке необходимо выбрать нужный нам SDK, в нашем случае Android 2.3.3 (API 10). Если телефона нет, то выбирайте 2.3.3 или выше; а если есть — версию, совпадающую с версией ОС телефона. Затем нажимаем на кнопку «Install Packages», чтобы запустить процесс установки.



После завершения скачивания и установки мы начинаем создавать приложение. Выбираем «File → New → Android Application Project». Заполним содержимое окна так, как показано на рисунке.



- Application Name — то имя приложения, которое будет показываться в Google Play Store. Но выкладывать приложение мы не собираемся, поэтому имя нам не особо важно.
- Project Name — имя проекта в ADT.

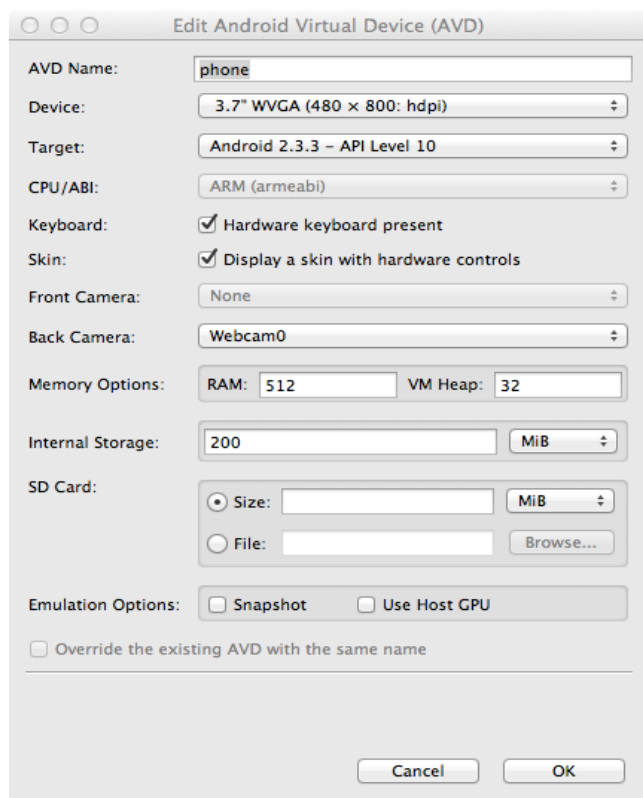
- Package Name — идентификатор приложения. Он должен быть составлен следующим образом: название Вашего сайта задом наперед, плюс какое-либо название приложения.

В выпадающих списках «Minimum Required SDK», «Target SDK», «Compile With» выбираем ту версию, которую мы скачали ранее. Более новые версии SDK поддерживают графические темы для приложений, а старые нет. Поэтому в поле «Theme» выбираем «None». Нажимаем «Next». Снимаем галочку с «Create custom launcher icon»: в рамках данной статьи не будем заострять внимание на создании иконки приложения. Нажимаем «Next». В появившемся окне можно выбрать вид «Activity»: вид того, что будет на экране, когда будет запущено приложение. Выбираем «Blank activity», что означает, что мы хотим начать всё с чистого листа. Нажимаем «Next». В нашем приложении будет всего одно Activity, поэтому в появившемся окне можно ничего не менять. Поэтому просто жмем на «Finish».

Все, наше приложение создано.

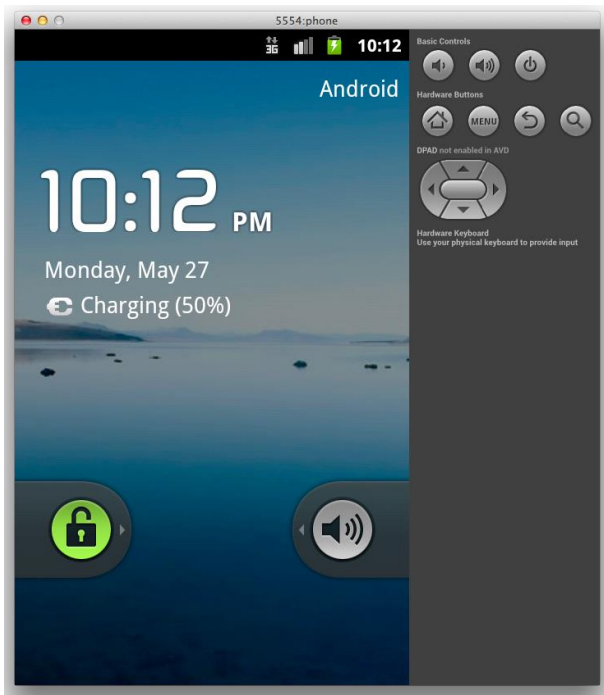
Настройка эмулятора

Отладка приложений для Android производится на реальном устройстве или, если такового нет, то на эмуляторе. Сконфигурируем свой. Для этого запустим «Window → Android Virtual Device Manager». В появившемся окне нажмем «New». Заполняем поля появившейся формы. От них зависит сколько и каких ресурсов будет предоставлять эмулятор «телефону». Выберите разумные значения и нажимайте «OK».



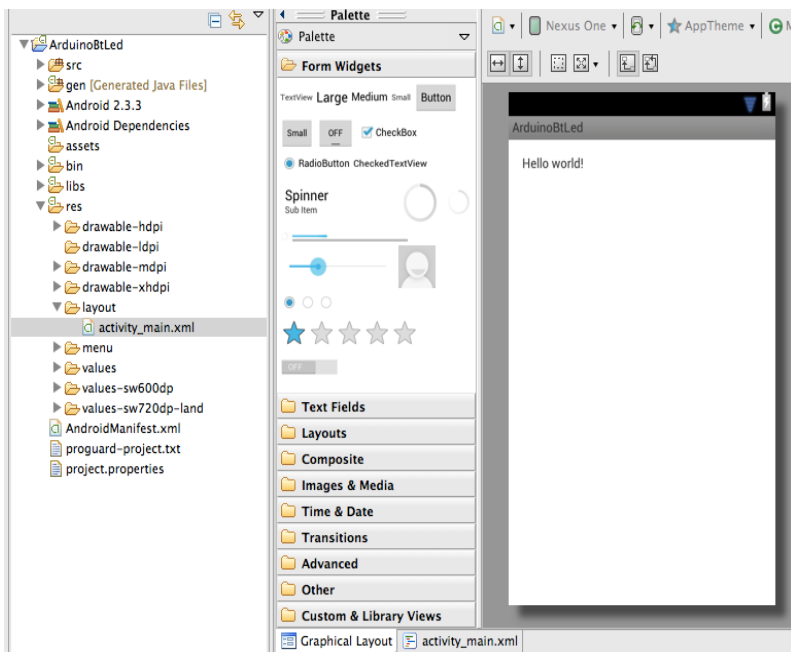
В окне Android Virtual Device Manager нажимаем кнопку «Start». Это запустит эмулятор. Запуск занимает несколько минут. Так что наберитесь терпения.

В результате вы увидите окно эмулятора подобное этому:



Заполнение Activity

Activity — это то, что отображается на экране телефона после запуска приложения. На нем у нас будет две кнопки «Зажечь красный светодиод» и «Зажечь синий светодиод». Добавим их. В панели «Package Explorer» открываем `res/layout/activity_main.xml`. Его вид будет примерно таким же, как на скриншоте.



Перетаскиваем 2 кнопки «ToggleButton» на экранную форму. Переключаемся во вкладку «activity_main.xml» и видим следующий код:

activity_main_automogen.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
```

```
<ToggleButton
    android:id="@+id/toggleButton1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:text="ToggleButton" />
```

```
<ToggleButton
    android:id="@+id/toggleButton2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView1"
    android:layout_alignParentRight="true"
    android:text="ToggleButton" />
```

```
</RelativeLayout>
```

Это ни что иное, как наша Activity, которая отображается не в виде графики, а описанная в формате XML.

Сделаем имена компонентов более понятными. Изменим поля `android:id` следующим образом.

```
<ToggleButton
    android:id="@+id/toggleRedLed" ...
```

```
<ToggleButton
    android:id="@+id/toggleGreenLed" ...
```

А еще добавим им подписи, изменим их цвет и размер текста. Результирующий код разметки будет выглядеть следующим образом.

activity_main.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
```



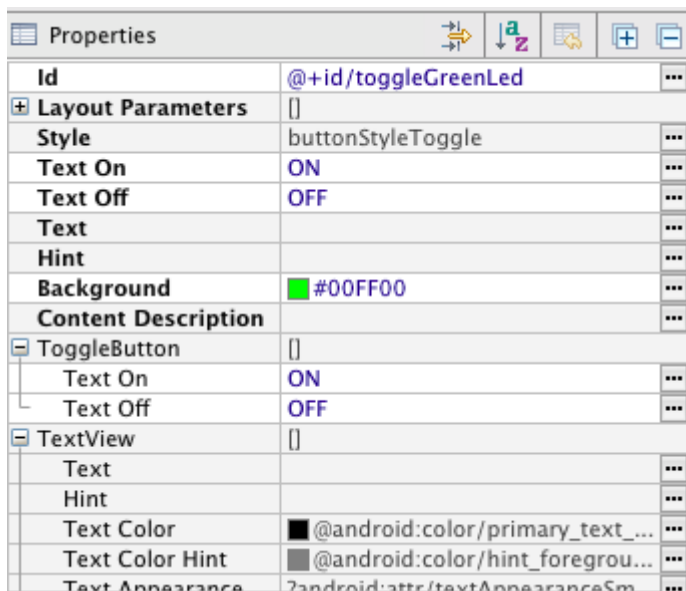
```
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity"
android:weightSum="2"
android:orientation="horizontal">
```

```
<ToggleButton
    android:id="@+id/toggleRedLed"
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:background="#FF0000"
    android:textOff="OFF"
    android:textOn="ON"
    android:textSize="30dp" />
```

```
<ToggleButton
    android:id="@+id/toggleGreenLed"
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:background="#00FF00"
    android:textOff="OFF"
    android:textSize="30dp"
    android:textOn="ON" />
```

```
</LinearLayout>
```

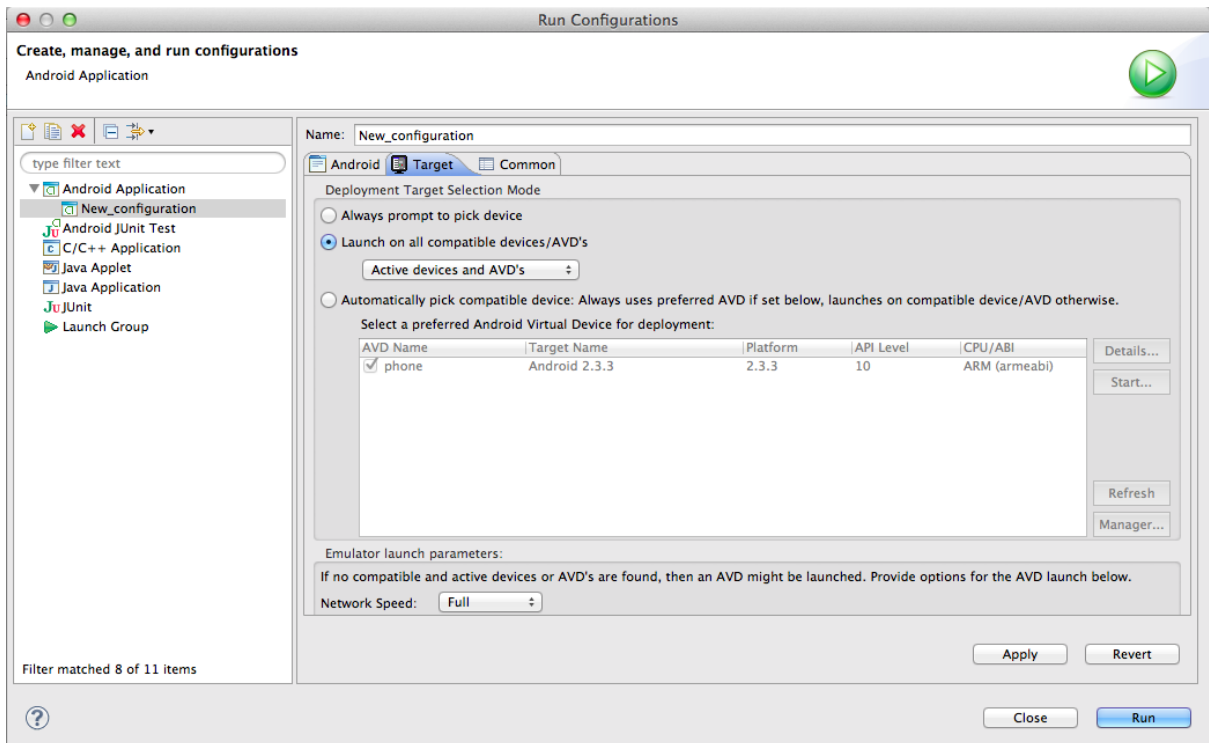
Эти же изменения можно сделать и в графическом режиме, воспользовавшись вкладкой «Outline/Properties».



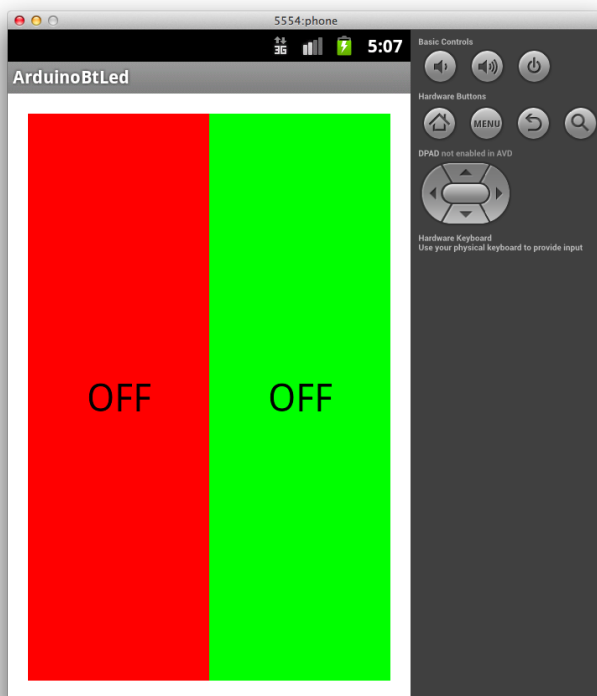
Пробный запуск

Мы можем запустить только что созданное приложение на эмуляторе. Идем в настройки запуска «Run» → Run Configurations», в левой части нажимаем на «Android Application». Появляется новая конфигурация «New_configuration».

В правой части окна выбираем вкладку «Target» и выбираем опцию «Launch on all compatible devices/AVD».



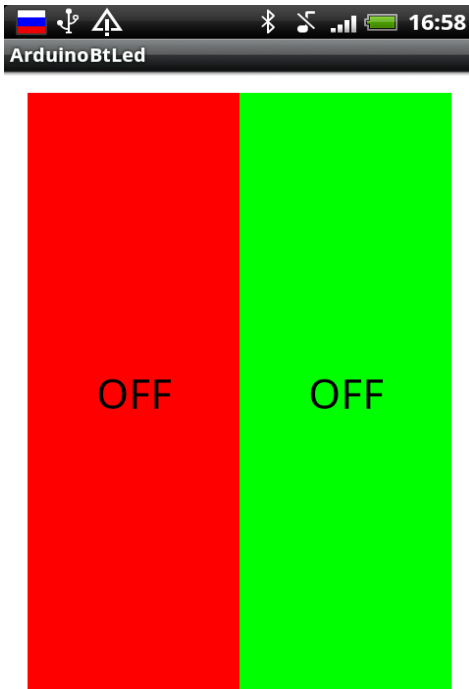
Нажимаем «Apply», а затем «Run». Приложение запустится в эмуляторе.



Можно понажимать кнопки. Но ничего происходить не будет, поскольку обработчики нажатий еще нами не написаны.

Чтобы запустить приложение на реальном устройстве, необходимо включить в его настройках опцию «Отладка USB» и подключить его к компьютеру.

На реальном устройстве приложение выглядит абсолютно аналогично.



Написание кода для Android

Правка манифеста

Каждое Android-приложение должно сообщить системе о том, какие права необходимо ему предоставить. Перечисление прав идет в так называемом файле манифеста `AndroidManifest.xml`. В нем мы должны указать тот факт, что хотим использовать Bluetooth в своем приложении. Для этого достаточно добавить буквально пару строк:

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="ru.amperka.arduinoobtled"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="10"
        android:targetSdkVersion="10" />

    <!-- Разрешаем приложению работать с Bluetooth -->
    <uses-permission android:name="android.permission.BLUETOOTH"/>
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
```

```

<activity
  android:name="ru.amperka.arduinoobtled.MainActivity"
  android:label="@string/app_name" >
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>

</application>

</manifest>

```

Добавляем основной код

Пришла пора вдохнуть жизнь в наше приложение. Открываем файл `MainActivity.java` (src → ru.amperka.arduinoobtled). Изначально он содержит следующий код:

MainActivityAutogen.java

```
package ru.amperka.arduinoobtled;
```

```
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
```

```
public class MainActivity extends Activity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
```

```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
```

```
}
```

Дополним код в соответствии с тем, что нам нужно:

1. Будем включать Bluetooth, если он выключен.
2. Будем обрабатывать нажатия на кнопки
3. Будем посылать информацию о том, какая кнопка была нажата.

Передавать на Arduino мы будем один байт с двузначным числом. Первая цифра числа — номер пина, к которому подключен тот или иной светодиод, вторая — состояние светодиода: 1 — включен, 0 — выключен.

Число-команда, рассчитывается очень просто: Если нажата красная кнопка, то берется число 60 (для красного светодиода мы выбрали 6-й пин Arduino) и к нему прибавляется 1 или 0 в зависимости от того, должен ли сейчас гореть светодиод или нет. Для зеленой кнопки всё аналогично, только вместо 60 берется 70 (поскольку зеленый светодиод подключен к 7 пину). В итоге, в нашем случае, возможны 4 команды: 60, 61, 70, 71.

Напишем код, который реализует всё сказанное.

MainActivity.java

```
package ru.amperka.arduinoobtled;
```

```
import java.io.IOException;
import java.io.OutputStream;
import java.lang.reflect.InvocationTargetException;
import java.lang.reflect.Method;
```

```
import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Toast;
import android.widget.ToggleButton;
```

```
public class MainActivity extends Activity implements View.OnClickListener{
```

```
    //Экземпляры классов наших кнопок
    ToggleButton redButton;
    ToggleButton greenButton;
```

```
    //Сокет, с помощью которого мы будем отправлять данные на Arduino
    BluetoothSocket clientSocket;
```

```
    //Эта функция запускается автоматически при запуске приложения
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
        //"Соединям" вид кнопки в окне приложения с реализацией
        redButton = (ToggleButton) findViewById(R.id.toggleRedLed);
        greenButton = (ToggleButton) findViewById(R.id.toggleGreenLed);
```

```
        //Добавлем "слушатель нажатий" к кнопке
        redButton.setOnClickListener(this);
        greenButton.setOnClickListener(this);
```

```

//Включаем bluetooth. Если он уже включен, то ничего не произойдет
String enableBT = BluetoothAdapter.ACTION_REQUEST_ENABLE;
startActivityForResult(new Intent(enableBT), 0);

//Мы хотим использовать тот bluetooth-адаптер, который задается по умолчанию
BluetoothAdapter bluetooth = BluetoothAdapter.getDefaultAdapter();

//Пытаемся проделать эти действия
try{
    //Устройство с данным адресом - наш Bluetooth Bee
    //Адрес определяется следующим образом: установите соединение
    //между ПК и модулем (пин: 1234), а затем посмотрите в настройках
    //соединения адрес модуля. Скорее всего он будет аналогичным.
    BluetoothDevice device = bluetooth.getRemoteDevice("00:13:02:01:00:09");

    //Иницилируем соединение с устройством
    Method m = device.getClass().getMethod(
        "createRfcommSocket", new Class[] {int.class});

    clientSocket = (BluetoothSocket) m.invoke(device, 1);
    clientSocket.connect();

    //В случае появления любых ошибок, выводим в лог сообщение
} catch (IOException e) {
    Log.d("BLUETOOTH", e.getMessage());
} catch (SecurityException e) {
    Log.d("BLUETOOTH", e.getMessage());
} catch (NoSuchMethodException e) {
    Log.d("BLUETOOTH", e.getMessage());
} catch (IllegalArgumentException e) {
    Log.d("BLUETOOTH", e.getMessage());
} catch (IllegalAccessException e) {
    Log.d("BLUETOOTH", e.getMessage());
} catch (InvocationTargetException e) {
    Log.d("BLUETOOTH", e.getMessage());
}

//Выводим сообщение об успешном подключении
Toast.makeText(getApplicationContext(), "CONNECTED",
Toast.LENGTH_LONG).show();

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

//Как раз эта функция и будет вызываться

```

```

@Override
public void onClick(View v) {

    //Пытаемся послать данные
    try {
        //Получаем выходной поток для передачи данных
        OutputStream outputStream = clientSocket.getOutputStream();

        int value = 0;

        //В зависимости от того, какая кнопка была нажата,
        //изменяем данные для отправки
        if (v == redButton) {
            value = (redButton.isChecked() ? 1 : 0) + 60;
        } else if (v == greenButton) {
            value = (greenButton.isChecked() ? 1 : 0) + 70;
        }

        //Пишем данные в выходной поток
        outputStream.write(value);

    } catch (IOException e) {
        //Если есть ошибки, выводим их в лог
        Log.d("BLUETOOTH", e.getMessage());
    }
}
}

```

Написание скетча

Данные, которые принимает Bluetooth-модуль, приходят через UART (он же Serial-соединение) на скорости 9600 бит/с. Настраивать Bluetooth-модуль нет никакой необходимости: он сразу готов к работе. Поэтому скетч должен уметь следующее:

1. Принять данные по UART
2. Зажечь нужный светодиод в зависимости от принятого кода

bluetooth.ino

```

void setup() {
    //Устанавливаем скорость UART
    Serial.begin(9600);

    //Настраиваем нужные пины на выход
    pinMode(6, OUTPUT);
    pinMode(7, OUTPUT);
}

void loop() {
    //Если данные пришли
    if (Serial.available() > 0) {

```



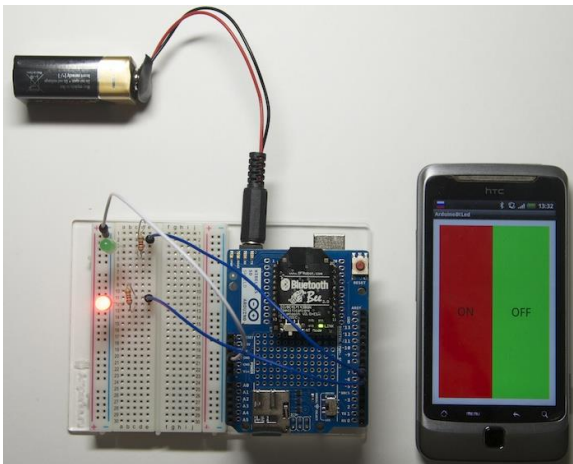
```
//Считываем пришедший байт
byte incomingByte = Serial.read();

//Получаем номер пина путем целочисленного деления значения принятого байта на
10
//и нужное нам действие за счет получения остатка от деления на 2:
//(1 - зажечь, 0 - погасить)
digitalWrite(incomingByte / 10, incomingByte % 2);
}
}
```

Особенности заливки скетча

Для связи Bluetooth-Веe с контроллером используются те же пины (0 и 1), что и для прошивки. Поэтому при программировании контроллера переключатель «SERIAL SELECT» на «Wireless Shield» должен быть установлен в положение «USB», а после прошивки его надо вернуть в положение «MICRO».

Результат



Заключение

В практической части мы научились создавать приложения для операционной системы Android и передавать данные по Bluetooth. Теперь при нажатии на кнопку на экране телефона на базе операционной системы Android, произойдет изменение состояния светодиода на плате.

В заключении, Wi-Fi — это современная технология беспроводной связи, которая является мобильной и практичной, но ее защищенность оставляет желать лучшего.

Из вышесказанного можно заключить, что WEP — устаревший протокол защиты беспроводного соединения. Рекомендуется не использовать WEP, если циркулирующая информация в сети имеет коммерческую важность. Говоря о протоколе WPA, пришедшему на замену WEP протоколу, следует сказать, что его плюсами являются усиленная безопасность данных и усиленный контроль доступа к беспроводным сетям. Но в практическом примере реализации атаки на протокол WPA видно, что протокол WPA, так же, как и WEP, имеет ряд недостатков. Для безопасного использования протокола WPA необходимо при выборе пароля использовать слова, не имеющие смысла (axdrtyh5nuo275bgdds — случайную или псевдослучайную последовательность символов), используя такие слова, вероятность успешного выполнения словарной атаки сводится к нулю. Для создания надёжной системы безопасности беспроводных сетей разработано немало методов. К примеру, самым надёжным способом считается использование виртуальных частных сетей VPN (Virtual Private Network).

Технология Bluetooth в настоящее время удовлетворяет всем основным требованиям, предъявляемым к беспроводной передаче данных даже в сложных условиях промышленного производства. Массовое применение привело к тому, что сегодня - это самая дешевая технология. Модули Bluetooth обеспечивают надежную передачу информации в условиях высокого уровня электромагнитного излучения различной природы, имеют низкую цену, невысокое энергопотребление, просты в использовании. Они позволяют получать удаленный доступ к устройствам и механизмам, проводная связь с которыми невозможна или затруднена. Bluetooth уже встроен во многие современные устройства — ноутбуки, КПК, телефоны, и имеет встроенную поддержку в наиболее распространенных операционных системах. Это дает возможность легко интегрировать данную технологию в создаваемые беспроводные системы управления и сбора информации.

Список литературы:

1. Федоров В. Модули Bluetooth в промышленных приложениях и системах сбора информации. - Москва: Электронное издание «Компоненты и технологии», 2007. - №
2. Попов М. Применение Bluetooth в радиоловительских устройствах // Радио. - 2010. - № 6. - 64 с.
3. Курт Ф. Bluetooth: решения на все или почти все случаи жизни. - Москва: Электронное издание «Компоненты и технологии». - 2004. - № 3.
4. А. Рестович, И. Стоян, И. Чубич. Bluetooth технология беспроводной связи и ее применение. - Загреб, Хорватия: Эрикссон Никола Тесла, 2005 – 86 с.
5. Архипкин В.Я., Архипкин А.В. Bluetooth технические требования практическая реализация приложения. - Москва: Мобильные коммуникации, 2002 – 203 с.
6. Лепихин И. Существует ли альтернатива Bluetooth?. - Москва: Компоненты и технологии, 2003. -№ 7.
7. Кратько А., Александров Р. Обзор компонентов Bluetooth. - Москва: Электронное издание «Компоненты и технологии. - 2005. - № 9.
8. Финогеев А.Г. Беспроводные технологии передачи данных для создания систем управления и персональной информационной поддержки // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы». - 2008. - 51 с.

Ссылки:

1. <http://articlekz.com/article/13574>
2. <http://wiki.amperka.ru/%D0%B1%D0%B5%D1%81%D0%BF%D1%80%D0%BE%D0%B2%D0%BE%D0%B4%D0%BD%D0%B0%D1%8F-%D1%81%D0%B2%D1%8F%D0%B7%D1%8C:android-%D0%B8-bluetooth>